Cluster-Based Point Cloud Analysis for Rapid Scene Interpretation

Eric Wahl[1] and Gerd Hirzinger[1]

Institute of Robotics and Mechatronics,
German Aerospace Center (DLR),
eric.wahl@dlr.de

Abstract. A histogram-based method for the interpretation of three-dimensional (3D) point clouds is introduced, where point clouds represent the surface of a scene of multiple objects and background. The proposed approach relies on a pose-invariant object representation that describes the distribution of surface point-pair relations as a model histogram. The models of the used objects are previously trained and stored in a database. The paper introduces an algorithm that divides a large number of randomly drawn surface points, into sets of potential candidates for each object model. Then clusters are established in every model-specific point set. Each cluster contains a local subset of points, which is evaluated in six refinement steps. In the refinement steps point-pairs are built and the distribution of their relationships is used to select and merge reliable clusters or to delete them in the case of uncertainty. In the end, the algorithm provides local subsets of surface points, labeled as an object. In the experimental section the approach shows the capability for scene interpretation in terms of high classification rates and fast processing times for both synthetic and real data.

1 Introduction

Scene interpretation is an important basic step for several different applications. Therefore, it is most desirable to have a generic approach that fits to every kind of three-dimensional (3D) object and that minimizes application-specific adaption as much as possible.

Statistics of geometric feature distributions of shape are a generic description of free-form objects. Examples of representations for one and two-dimensional feature distributions are discussed in [1, 2]. These approaches are capable of rapidly classifying objects with respect to large databases. However, the objects need to be isolated from background and suffer from moderate recognition accuracy, so that they are more suitable for only a coarse preliminary search.

An alternative line of research addresses local shape distributions that allow scenes of multiple objects and backgrounds. The *spin images* of Johnson and Hebert [3] as well as the *surface signatures* of Yamany and Farag [4] contributed a lot to local distribution analysis in cluttered scenes. To create their histograms, surface points are picked and a plane is rotated about their local surface normals. The surrounding points are accumulated in that plane. Thus, both approaches require dense surface meshes and relatively large memory consumption for object representation.

This paper relies on a generic histogram-based description of shape as introduced in [5]. The object representation shows good performance with respect to recognition rate, processing time, memory consumption, and descriptive capacity. The contribution of this work is an expansion of the formerly isolated free-form objects to scenes that include several objects and background. A new cluster-based method is introduced, which iteratively separates object points from the point cloud with respect to the local distribution of point-pair relations.

2   Object Representation and Training Phase

This section summarizes the object representation (see [5] for details) and expands the former description with two additional parameters.

Free-form shapes that are composed of oriented surface points can be described by the statistical distribution of point-pair relations. An oriented point consists of its position $\mathbf{u}$ and its local surface normal $\mathbf{v}$ and is referred to as a *surflet* in the following.

The relationship of a surflet $\mathbf{p_1} = (\mathbf{u_1}, \mathbf{v_1})$ to another surflet $\mathbf{p_2} = (\mathbf{u_2}, \mathbf{v_2})$ is uniquely encoded by four parameters α, β, γ, and δ. The attributes α and β respectively represent $\mathbf{v_2}$ as an azimuthal angle and the cosine of a polar angle, with respect to $\mathbf{p_1}$. The parameters γ and δ represent the direction and length of the translation from $\mathbf{u_1}$ to $\mathbf{u_2}$, respectively. Furthermore, we refer to the relationships of a surflet-pair $\mathbf{s} = (\mathbf{p_1}, \mathbf{p_2})$ as a *feature* $\mathbf{f_s} = (\alpha, \beta, \gamma, \delta)$.

Taking all possible features into account leads to a characteristic distribution of an object surface. The feature distribution is quantized in a histogram H_M of normalized frequencies. For each dimension we use five quantization steps. This produces a number of $d = 5^4 = 625$ bins per histogram. The function $h_M : \mathbf{f_s} \mapsto \mathbf{b} \in \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{d}\}$ maps a feature to the corresponding bin $H_M(b)$.

A model $M = (H_M, \delta_{\mathrm{M,max}}, \tilde{r}_M, r_{\mathrm{M,max}})$ consists of a normalized histogram of feature probabilities and three characteristic parameters. The first parameter is the maximum distance $\delta_{\mathrm{M,max}}$ between two object surflets. The last parameters are the mean distance $\tilde{r}_M$ and the maximum distance $r_{\mathrm{M,max}}$ of surface points to the centroid.

In the training phase, models are learned by uniformly drawing a large number of random surflet-pairs. The data source used for training can be either synthetical (e.g. from CAD models) or can be measurements from 3D-sensing. In the second case the measurements should contain no background surflets.

Experiments concerning the descriptive capacity of the object representation, i.e. recognition rates in the presence of noise, with partial visibility and variation of the surface point density, are discussed in [5].

3   Scene Interpretation

In this paper, we deal with 3D-scenes that consist of oriented surface points that build large point clouds.

Because we are representing an object as a distribution of its surflet-pair relations, the strategy of scene interpretation is to divide the point cloud into subsets and label them as an object or as background.

The structure of sensed data is often a point cloud with varying local densities, while objects are trained with randomly and uniformly drawn samples of the surface. The different conditions of surface density affects the probable location of a surflet and, therefore, lead to distorted feature histograms. To overcome this problem an initial reduction of data is necessary. A subdivision space (e.g. an octree or balanced binary tree, see [6] for details) is applied, which divides space into homogeneously-sized cells. Each cell contains the mean surflet value in its space. This preliminary step reduces the noise and size of data at once. Moreover, it guarantees an upper bound of surface density.

Since scenes consist of several objects and background, surflet-pairs are not necessary of one object only. Thus, the major problem the approach has to deal with, is to decide which surflet pairs can be trusted and which are *crosstalk*. Here, crosstalk is the relation of surflets that does not belong to the same source, e.g. a pair with one point being part of an object and one point being part of the background. The ratio of crosstalk to real object point-pairs contaminates the feature histograms and has to be kept as small as possible.

To reduce crosstalk resulting from background, two strategies are proposed. The first possibility is to *suppress* the background, in that case the position must be known and it must be the same for all measurements. A more flexible solution is to *identify and eliminate* the background. In this paper, we favor the second approach. To achieve this, the appearance of background features is trained like those of the objects.

In the following, an algorithm is proposed which cuts off local sets of points and classifies them with the label of an object. The processing is based on the previously reduced point cloud.

Step 1: Initialization of Cluster Seeds

As in the training phase, the algorithm starts with drawing a set $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ of n surflet-pairs $\mathbf{s}_i$. With regard to the k models of the database $\Omega = \{M_1, \ldots, M_k\}$, the maximal diameter $\delta_{M_j,\max}$ of the largest model M_j limits the distance allowed in a surflet-pair.

Subsequently, the features $\mathbf{f_s}$ are calculated. At this stage it is not known whether both surflets are part of the same object or rather describe a false relation, that is crosstalk. Consequently, a second reduction step has to be applied which determines the sample set

$$\mathcal{S}_{M_i}^* = \{\mathbf{p_s} \mid \exists\, \mathbf{s} \in \mathcal{S} : \mathbf{p_s} \in \mathbf{s} \quad \wedge$$
$$H_{M_i}(h_{M_i}(\mathbf{f_s})) \geq \tilde{H}_{M_i} \quad \wedge \tag{1}$$
$$[H_{M_i}(h_{M_i}(\mathbf{f_s})) \geq H_{M_j}(h_{M_j}(\mathbf{f_s}))\forall j \neq i]\}, \tag{2}$$

for each model M_i individually. The threshold

$$\tilde{H}_M = \frac{\lambda_H}{m} \sum_{\forall \mathbf{f_s}} H_M(h_M(\mathbf{f_s})), \text{where } m = \mathrm{card}\{\mathbf{f_s} | H_M(h_M(\mathbf{f_s})) \neq 0\}, \tag{3}$$

restricts $\mathcal{S}_{M_i}^*$ to the significant surflets $\mathbf{p_s}$ of the model M_i. It can be adjusted by a positive scalar[1] λ_H. The notation card is used for the cardinality of the set. Condition 2 limits the set to features, which respond most for the preferred model. Surflets collected by the background model need no further processing and affect classification no longer.

It is assumed that surflets in $\mathcal{S}_{M_i}^*$ build spatial clusters for objects represented by the model M_i. Accordingly, m random surflets are drawn therefrom. These surflets initialize the positions of a model-dependent set $\mathcal{C}_{M_i} \subseteq \mathcal{S}_{M_i}^*$ of *cluster seeds* $\mathbf{c}$. The term "cluster seed" is used to emphasize the preliminary state as compared to a cluster.

Step 2: Cluster Seed Refinement

Each cluster seed $\mathbf{c} \in \mathcal{C}_{M_i}$ has a sphere of the radius $r_{M_i,\max}$ and initially contains the local subset

$$\mathcal{S}'_{\mathbf{c},M_i} = \{\mathbf{p_s} \in \mathcal{S}_{M_i}^* \mid r_{M_i,max} \geq \|\mathbf{p_s} - \mathbf{c}\|\} \subseteq \mathcal{S}_{M_i}^*. \tag{4}$$

Not all of these surflets are reliable, so two alternating steps are iteratively applied for refinement.

Regrouping: Assuming that all surflets belong to the same object, a recombination of the surflets should produce acceptable features. On the other hand, false surflets that previously built also a valid feature by chance are now likely to cause crosstalk. The features of the recombined surflets are calculated and compared to the histogram H_{M_i} of the model. Impossible features that point to a zero value ($H_{M_i}(h_{M_i}(\mathbf{f_s})) = 0$) are labeled as crosstalk.

Crosstalk Elimination: A cluster seed is valid if it contains no crosstalk. All surflets that cause crosstalk are erased.

Both steps are applied until no crosstalk occurs or no surflets remain. Cluster seeds without surflets are deleted. The surflet sets of the remaining cluster seeds are referred to as $\mathcal{S}''_{\mathbf{c},M_i} \subset \mathcal{S}'_{\mathbf{c},M_i}$.

Step 3: Cluster Seed Merging

Due to the large number of initial cluster seeds in relation to the number of expected objects, many cluster seeds belong to the same object. Cluster seeds of the same object share a large number of surflets. It is obvious that such cluster seeds should be merged into one large *cluster*.

To achieve good merging results it is advisable to start with the best cluster seeds. Therefore, a criterion is needed to rate the distribution quality of each cluster seed and its surflet set $\mathcal{S}''_{\mathbf{c},M_i}$. In [7] and [5] it is shown that the logarithmic likelihood outperforms other criteria like Kullback-Leibler divergence and χ^2-test in processing time and reliability. Accordingly, the criterion

$$\mathcal{L}(M_i|\mathcal{S}''_{\mathbf{c},M_i}) = \frac{1}{\mathrm{card}\mathcal{S}'_{\mathbf{c},M_i}} \sum_{\mathbf{p_s} \in \mathcal{S}'_{\mathbf{c},\mathbf{M_i}}} \ln H_{M_i}(h(\mathbf{f_s})) \tag{5}$$

---

[1] We used $\lambda_H = 1.3$ in the tests.

is applied.

Since higher votes $\mathcal{L}(M_i|\mathcal{S}''_{\mathbf{c},M_i})$ refer to better model fits and a larger number of valid surflets, we can use $\mathcal{L}(M_i|\mathcal{S}''_{\mathbf{c},M_i})$ to sort the cluster seeds by their quality.

Subsequently, the best ranked cluster seed is compared to all lower-ranked seeds. Each time the number of coincident surflets exceeds a certain threshold[2] λ_o, the common surflets are stored in a cluster $\hat{\mathbf{c}}$. Restriction to surflets contained in both merging cluster seeds, reduces undetected crosstalk. Such crosstalk occurs when a cluster seed is placed in between two objects of the same model. Then, half of the set may point to one object and the other half to the second object.

Merged cluster seeds are deleted from the sorted list. If a cluster seed remains, the algorithm starts again at the top of the list. Otherwise, the merging process is finished.

Step 4: Distribution Checkup

After merging, the set of surflets has to be checked again. As previously done with the cluster seeds, the surflets of a cluster are regrouped and the features as well as their distribution histogram are calculated. Clusters of this state should contain nearly no crosstalk, thus we additionally demand the highest distribution response when applying (5) on the expected model. Otherwise the cluster is deleted.

Step 5: Cluster Similarity

Only few clusters $\hat{\mathbf{c}} \in \hat{\mathcal{C}}_M = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_\mathbf{m}\}$ reach this level. Misclassifications are rare but still possible, since similar parts of different objects may show similar distributions, e.g. the bottom of a cup and the bottom of a bottle. Nevertheless, the size of a cluster contains hints to the correctness of classification that could be described by two conditions. The first condition is that the cluster size has to exceed a minimum. Second condition is that all clusters of a model type should be of similar size. The size is measured in terms of the number $p_{\hat{\mathbf{c}}}$ of contained surflets, where $p_{\max}$ is the surflet number of the largest cluster. The normalized surflet number $p_{\hat{c}}$ of each accepted cluster $\hat{c}$ must be in the interval $[\lambda_{\hat{c}}, 1]$, where $\lambda_{\hat{c}}$ is a threshold for the minimum number of surflets. Division by $p_{\max}$ is used for normalization.

Step 6: Cluster Competition

A last evaluation step prevents multiple classifications of an object. Therefore the centroid $\mathbf{u}_{\hat{c}_1}$ of a cluster $\hat{c}_1$ is calculated and compared to the centroids of all other clusters, e.g. the centroid $\mathbf{u}_{\hat{c}_2}$ of cluster $\hat{c}_2$. A collision occurs, if

$$\| \mathbf{u}_{\hat{c}_1} - \mathbf{u}_{\hat{c}_2} \| \leq \max(\tilde{r}_{M_i,\hat{c}_1}, \tilde{r}_{M_j,\hat{c}_2}), \tag{6}$$

holds, where $\tilde{r}_{M_i,\hat{c}_1}$ is the mean distance of the surflets to the centroid $\mathbf{u}_{M_j\hat{c}_1}$ (see Section 2). Two cases have to be distinguished. In the first case, both clusters are of the same model type ($M_i = M_j$) and then can be unified. In the second type, the clusters

---

[2] 30% of co-occurrence shows good performance.

are of different model type ($M_i \neq M_j$). Thus, the cluster $\hat{c}_1$ of model M_i remains, if it fulfills at least two of the three following conditions:

$$p_{\hat{c}_1} > p_{\hat{c}_2}, \tag{7}$$

$$p_{\hat{c}_1}/n_{\hat{c}_1} > p_{\hat{c}_2}/n_{\hat{c}_2}, \tag{8}$$

$$\text{and} \quad n_{\hat{c}_1}/n_{\max,\hat{c}_1} > n_{\hat{c}_2}/n_{\max,\hat{c}_2}. \tag{9}$$

Condition 7 favors the cluster with the larger number of surflets with respect to its model-specific ranking, which was already calculated in Step 4. Condition 8 considers the number of surflets $p_{\hat{c}}$ in relation to the number $n_{\hat{c}}$ of merged cluster seeds per cluster. Finally, Condition 9 addresses the number of merged cluster seeds $n_{\hat{c}}$ in relation to the maximum number $n_{\max,\hat{c}}$ of merged cluster seeds per model.

4 Experiments

The experiment section is split into two parts. The first part discusses results achieved with synthetic data, while the second part takes a closer look at results from real data.

4.1 Synthetic Data

In the case of synthetic data, we use a database of eight entries, which is a sufficiently large number for most applications in robotics. The database contains models for the surface of the table, the bunny, the bottle, the carafe, the cup, the dragon, the horse, and the wine glass. All objects were randomly placed on the plain in an upright position. The minimal distance from one object to every other is always $r_{M,\max}$ of the smaller one. Fig. 2(a) shows one of a 1000 examples used in this experiment. The mean number of scene points was about 770,000 surface points, which the octree structure reduced to 77,000 points, leading to the generation of 232,000 features.

The algorithm performed at a rate of 97.2% correctly classified objects. The horse showed the weakest results. This effect derives from the disadvantageous point distribution. The thin but long legs widen the object size and therefore increment the possibility of non-object points without adding enough points themselves.

The accuracy in terms of included cluster points being part of the object achieved 96.1%, where clusters contained a mean number of 330 points. Processing times on an Intel Xeon 2.8GHz processor with 2GB RAM averaged 6.8 seconds.

4.2 Real Data

The DLR multi-sensory 3D-Modeler (see [8] for details) was used for real data acquisition. We used the laser-range scanner seven times and the laser-profiler two times in sampling the 3D-scenes. In contrast to the tests on synthetic data, the real scenes exhibit noise and incomplete surfaces (see Fig. 2(b)). The reasons for this clutter are surface parts that are unreachable for the sensor or regions that produce no valid information due to local reflectance behavior of the surface.

Here, a database of four objects is used: a toy phone, a wooden train, a bust of Zeus, and a bottle. The algorithm managed to correctly classify all objects in all nine scenes. Due to a better ratio of object points to background points the number of drawn samples can be reduced. Thus, the processing times on an Intel Xeon 2.8GHz processor with 2GB RAM for real data was only about 1.57 seconds. The separated clusters consisted of about 495 surflets. The algorithm showed no different performance between laser-range scans and laser-profile scans.

In addition to the fast classification results, the algorithm showed minor sensitivity to the threshold values. There was no need to re-parameterize them in the case of real data, so the settings of the synthetic data could be used.

(a)   (b)   (c)   (d)
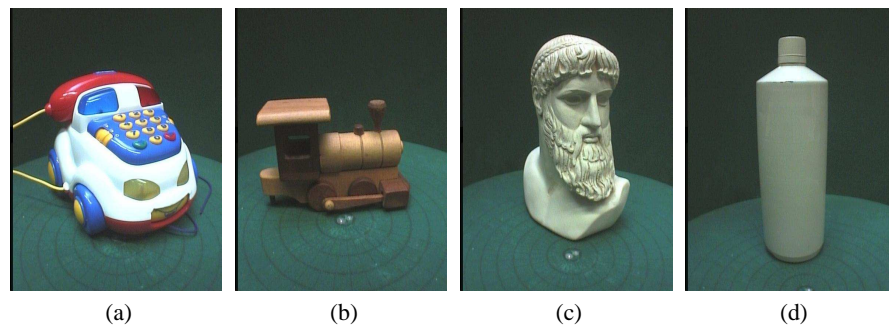
Fig. 1. The real objects database: (a) a toy phone, (b) a wooden train, (c) a bust of Zeus, and (d) a bottle;

(a)   (b)   (c)
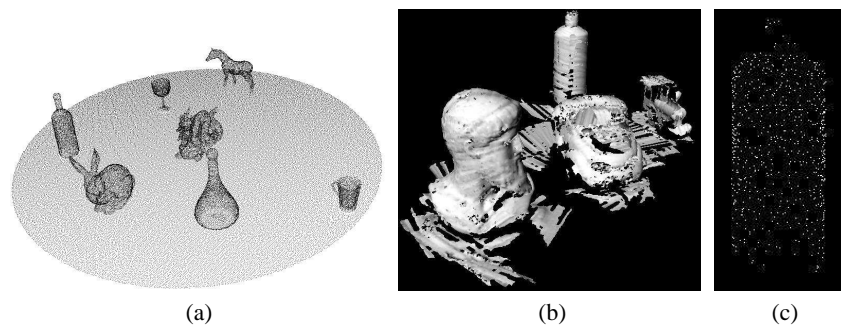
Fig. 2. (a) An example of all trained objects arbitrarily placed on a table. The scene shows the point cloud after the initial data reduction. (b) The surface mesh calculated from a 3D laser-scan of a scene. Holes on the surface result from unreachable regions or disadvantageous surface conditions (color, transparency, etc.). (c) The remaining surface points of the correctly classified bottle.

5   Conclusion

In this paper, we introduced a method for interpretation of 3D point cloud scenes using a previously trained database. A generic free-form description is used, which models the shape of an object by the statistical distribution histogram of parameterized four-dimensional relations of surface point-pairs.

Note that we are using a pose-invariant description. Only the point clouds can be extracted as input for an *iterative-closest-point algorithm (ICP)*, for example, to acquire object pose if necessary.

The proposed algorithm localizes objects in the scene by initially drawing a large number of random surface points and grouping them into pairs. Afterwards, the point-pair relations are calculated and the most likely origin for them are investigated. This is done in six evaluation steps by first establishing preliminary clusters that are later merged into final clusters if they are deemed trustworthy or that are deleted elsewhere. Each cluster consists of a set of local points. Every evaluation step is a refinement of the sets, such that the result of the algorithm provides sets of surface points that are labeled with one of the trained objects.

In the experimental section we showed that our approach is capable of fast classification of objects in the presence of background. The experiments addressed classification results and processing times for both synthetic and real data. The generic free-form description, which allows training by CAD models or 3D-scans, combined with the high classification rates (above 97%) in less than 6.8 seconds for synthetic data and less than 2.1 seconds for real data, makes our algorithm an efficient tool for many applications in robotics.

References

1. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transcations on Graphics **21(4)** (2002) 807–832
2. Vandeborre, J.P., Couillet, V., Daoudi, M.: A practical aproach for 3D model indexing by combining local and global invariants. 3D Data Processing Visualization Transmission (3DPVT'02) (2002) 644–647
3. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 433–449
4. Yamany, S.M., Farag, A.A.: Surface signatures: An orientation indepentent free-form surface representation scheme for the purpose of objects registration and matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1105–1120
5. Wahl, E., Hillenbrand, U., Hirzinger, G.: Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification. International Conference on 3-D Digital Imaging and Modelling (2003)
6. Wahl, E., Hirzinger, G.: A method for fast search of variable regions on dynamic 3D point clouds. DAGM 2005 (27th Annual meeting of the German Association for Pattern Recognition) (2005)
7. Schiele, B., Crowley, J.L.: Recognition without correspondence using multidimensional receptive field histograms. International Journal of Computer Vision **36(1)** (2000) 31–52
8. Suppa, M., Hirzinger, G.: A novel system approach to multisensory data acquisition. The 8th Conference on Intelligent Autonomous Systems IAS-8 (2004)